A Brief Introduction to Deep Learning

Yingbo Zhou

University at Buffalo
The State University of New York

Overview

University at Buffalo
The State University of New York

Table of Contents

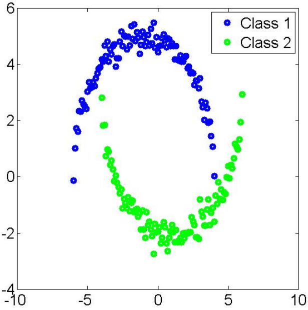
Machine Learning

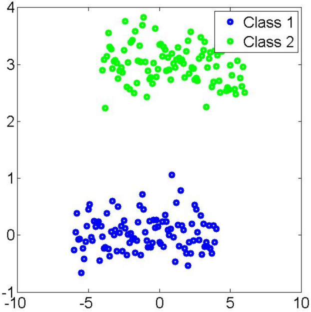
> "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E"

– Tom Mitchell, Machine Learning (1997)

Feature Representation

- Data (feature) representation is important





Feature Representation

- Data (feature) representation is important
- Feature engineering is inefficient
  - Features are problem dependent
  - Engineering require a lot of domain knowledge
  - Does not take advantage of the data
- Automatically learning data representation is desired
  - Deep learning is one of such method

Why Deep?

- Deep models are more efficient
  - There are functions that can be represented efficiently by using linear number of parameters while using a depth k structure, but will require exponential number more parameters for using one less layer
- Deep models are biologically plausible
  - Brain is a deep architecture
- Deep models provides an automated way of learning a hierarchical representation

What is Deep Learning?

- A re-brand of conventional multilayer neural networks (with some new tricks)
- Learning a hierarchy of data representations
- Learning non-linear transformations
- Is NOT how brain works!!
  - We do get inspirations from neural science though

Learning Deep Architecture

- Learning deep architecture is hard
  - Vanishing gradient
  - Objective is highly non-convex
- Greedy layerwise pre-training

  (Hinton and Osindero 2006, Bengio *et al.* 2006, Ranzato *et al.* 2006)
  - Decompose a deep network into multiple shallow ones
  - Using unsupervised method to train each layer

Greedy Layerwise Pre-training

An illustration

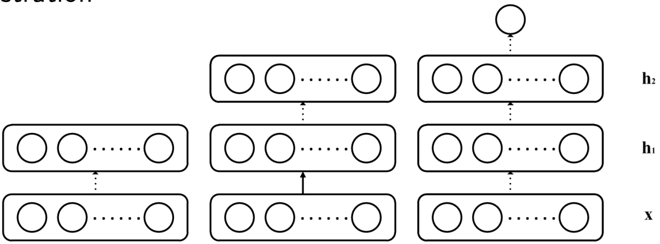


Table of Contents

Building Block for Deep Architecture

- Gradient-based Learning
- Layerwise pre-training
- Restricted Boltzmann machines
- Autoencoders
- Convolutional Networks

Energy Based Models

- Energy based models (EBMs)
  - Without latent variables
    $p(\mathbf{x}; \mathbf{\Theta}) = \frac{1}{Z(\mathbf{\Theta})} \exp\{-E(\mathbf{x}; \mathbf{\Theta})\}$
  - With latent variables
    $p(\mathbf{x}; \mathbf{\Theta}) = \frac{1}{Z(\mathbf{\Theta})} \exp\{-FE(\mathbf{x}; \mathbf{\Theta})\}$
    where $FE(\mathbf{x}; \mathbf{\Theta}) = -\log \sum_{\mathbf{h}} \exp\{-E(\mathbf{x}, \mathbf{h}; \mathbf{\Theta})\}$
- Maximum likelihood learning of EBMs

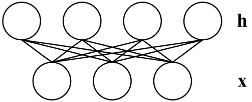
$$\mathbb{E}_{q(\mathbf{x})}[\frac{\partial \log p(\mathbf{x}; \mathbf{\Theta})}{\partial \mathbf{\Theta}}] = -\mathbb{E}_{q(\mathbf{x})}[\frac{\partial FE(\mathbf{x}; \mathbf{\Theta})}{\partial \mathbf{\Theta}}] + \mathbb{E}_{\tilde{\mathbf{x}} \sim p(\mathbf{x})}[\frac{\partial FE(\tilde{\mathbf{x}}; \mathbf{\Theta})}{\partial \mathbf{\Theta}}]$$

- Contrastive Divergence (Hinton 2002)
  - Using biased samples

University at Buffalo
The State University of New York

Restricted Boltzmann Machines

- Restricted Boltzmann Machines (Smolensky 1986)



$$E(\mathbf{x}, \mathbf{h}) = -\mathbf{a}^T\mathbf{x} - \mathbf{b}^T\mathbf{h} - \mathbf{h}^T W\mathbf{x}$$

$$p(\mathbf{h}|\mathbf{x}) = \prod_i p(h_i|\mathbf{x})$$

$$p(\mathbf{x}|\mathbf{h}) = \prod_j p(x_j|\mathbf{h})$$

$$p(h_i|\mathbf{x}) = s(\sum_j W_{ij}x_j + b_i)$$

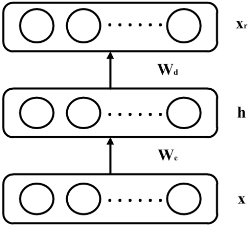
$$p(x_j|\mathbf{h}) = s(\sum_i W_{ij}h_i + a_j)$$

University at Buffalo
The State University of New York

Autoencoders

- Basic autoencoder (Hinton and Zemel 1994)

$$\mathbf{h} = f_e(\mathbf{x}) = s_e(\mathbf{W_e}\mathbf{x} + \mathbf{b_e})$$
$$\mathbf{x_r} = f_d(\mathbf{x}) = s_d(\mathbf{W_d}\mathbf{h} + \mathbf{b_d})$$

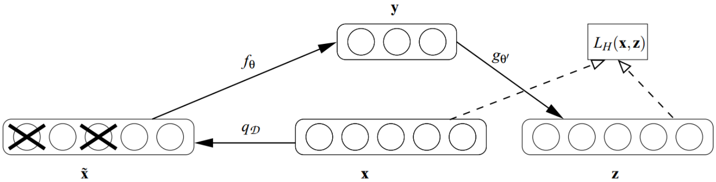


- Learning of autoencoders

$$\mathcal{J}_{AE}(\mathbf{\Theta}) = \sum_{\mathbf{x}\in\mathcal{D}} \mathcal{L}(\mathbf{x}, \mathbf{x_r})$$

Denoising Autoencoders

An illustration



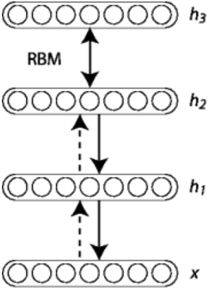
(Image from Vicent *et al.* 2010)

Table of Contents

University at Buffalo
The State University of New York

Deep Belief Networks

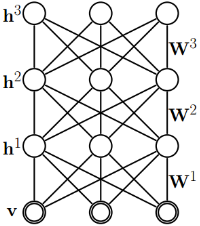
$$p(\mathbf{x}, \mathbf{h}^1, \ldots, \mathbf{h}^l) = p(\mathbf{h}^{l-1}, \mathbf{h}^l) \left(\prod_{k=1}^{l-2} p(\mathbf{h}^k | \mathbf{h}^{k+1}) \right) p(\mathbf{x}|\mathbf{h}^1)$$



Stacks of RBMs forms a deep belief network (DBN)

Deep Boltzmann Machines

$$E(\mathbf{v}, \mathbf{h}^1, \mathbf{h}^2, \mathbf{h}^3) = -\mathbf{v}^T \mathbf{W}^1 \mathbf{h}^1 - \mathbf{h}^{1T} \mathbf{W}^2 \mathbf{h}^2 - \mathbf{h}^{2T} \mathbf{W}^3 \mathbf{h}^3$$

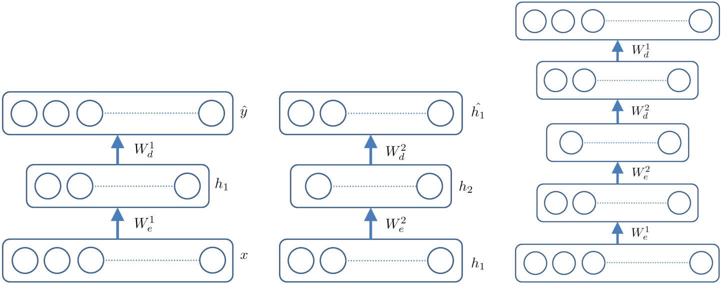


- Stacks of slightly modified RBMs can form a deep Boltzmann machine (DBM).
- Training is still challenging, can get easier with the proper initialization from RBMs.
- Can be trained from scratch using more recent techniques.

University at Buffalo
The State University of New York

Stacked Autoencoders

An illustration



Convolution Networks

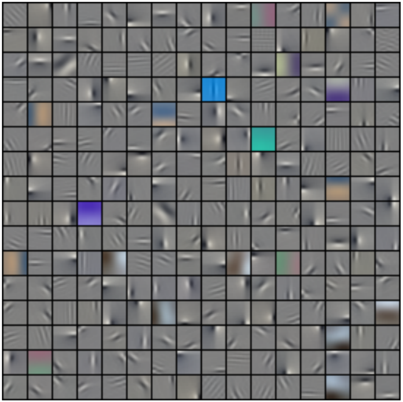
We will look at it next class

University at Buffalo
The State University of New York

Table of Contents

Examples

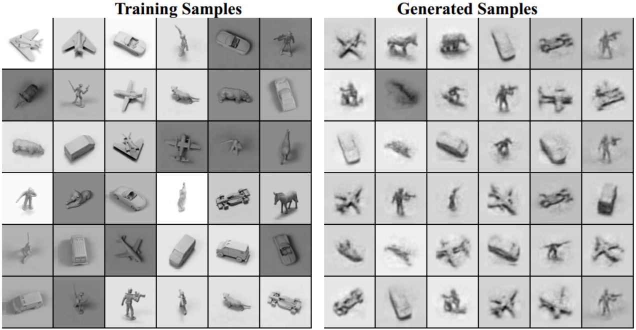
Filters learned from natural images



(Image from Ranzato *et al.* 2010)

Examples

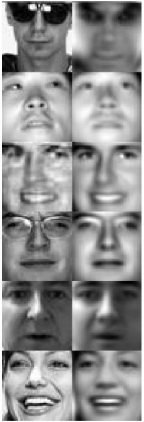
Samples from model



(Image from Salakhutdinov and Hinton 2009)

Examples

Reconstructions from model



Some Online Demos

Leading Research Groups

Geoffrey Hinton (Toronto)

Yann Lecun (NYU)

Youshua Bengio (UdeM)

Andrew Ng (Stanford)

Popular Software Packages

- Theano
- Pylearn2
- Cudaconvnet
- Torch

University at Buffalo
The State University of New York

Thanks

Questions?